(19) Japan Patent Office (JP)
(11) Patent Application Disclosure
(12) Unexamined Patent Application Publication (A) S63-217452

(51) Int. Cl.[4]	Identification No.	File No.
G 06 F 12/16	310	A-7737-5B
G 11 C 7/00	313	7341-5B
11/34	371	A-8522-5B
29/00	303	H-7737-5B

(43) Disclosure Date: September 9, 1988

No examination requested
Number of inventions: 1 (Total of 8 pages)

(54) Name of Invention: Method for Setting Memory Access Timing

(21) Patent Application S62-51509
(22) Application Date: March 6, 1987

(72) Inventor: Takashi Saito
Mitsubishi Electric Company, Ltd., Computer Manufacturing Facilities
325 Uemachiya, Kamakura City, Kanagawa Prefecture
(71) Applicant: Mitsubishi Electric Company, Ltd.
2-2-3 Marunouchi, Chiyoda-ku, Tokyo
(74) Representative: Masao Oiwa, Patent Attorney, and two others

Details

1. Name of Invention

Method for Setting Memory Access Timing

2. Patent Claims

A method for setting memory access timing in a logical circuit that accesses memory, characterized by the provision of a register for which a variety of values can be set by a program, by repetitively having the value set in said register be changed sequentially by the program and performing test writes to the memory and test reads from the memory where the data that is written is compared to the data that is read, by setting to the register the setting that was in effect when the results of the comparison matched, and by accessing the memory based on said setting.

3. Detailed Explanation of the Invention

[Area of Application in Industry]

This invention pertains to a method for setting the memory access timing in order to set the access timing for random access memory (hereinafter abbreviated "RAM") that is equipped in, for example, data processing devices.

[Prior Art]

Figure 6 shows a block diagram of the logic circuits that use the conventional memory access timing set method. In the figure, 1 is RAM (using the example where a dynamic RAM is used), 2 is an address multiplexer, 3 is a multiplexed address bus connecting the RAM 1 with the address multiplexer 2, 4 is an address bus connected to the address multiplexer 2, 5 is a data bus connected to the RAM 1, and 6 is a memory control ring. In addition, 7 is a flipflop for generating the row address select signal (RAS signal), hereinafter abbreviated "RAS flipflop," 8 is a flipflop for generating the column address select signal (CAS signal), hereinafter abbreviated "CAS flipflop," 9 is a flipflop for generating the column select signal (COLS signal), hereinafter abbreviated "COLS flipflop." 10 is an AND gate, 11, 12, and 13 are OR gates, 14, 15, and 16 are NOR gates, and 17, 18, 19, and 20 are jumper lines for selecting the output from the memory control ring 6 that is to be used. Additionally, for simplicity in the explanation, the logic circuits for refreshing the RAM 1 are not shown.

The operation of this method is described below. In this explanation, "1" indicates either the active level or the high logic level, while "0" indicates the inactive level or the low logic level. The memory control ring 6 is enabled and placed in an operational state when the memory access mode signal of line L1 goes to "1," and, synchronized to the master clock on line L2, the outputs T0, T1, . . . Tk, . . ., Tl, . . ., Tm, . . ., Tn, . . ., Te-1, Te are sequentially set to "1" in the state transitions. When the memory access mode signal is "0," all outputs T0 to Te from the memory control ring 6 go to "0." The respective flipflops 7, 8, and 9 each output their latched signals from the output terminal 1 on each, and output the inverse of the latched signal on output terminal 0 of each. The RAS signal, CAS signal, and WE signal applied, respectively, to the RAS, CAS, and WE terminals of RAM 1 are each active at "1." In addition, in this conventional example, jumper lines 17, 18, 19, and 20 are set by hand, selecting, respectively, outputs Tk, Tl, Tm, and Tn of the memory control ring 6.

Below will be explained an example of an operation to write to the RAM 1, referencing the timing chart shown in Figure 4. When the memory access commences, both the memory access mode signal on Line L1 and the write mode signal on Line 3 both go to "1." At this time, the address is applied to the address bus 4, the row address is selected by the address multiplexer 2 and is output on the multiplexed address bus 3. At this time the write data is applied to data bus 5.

In this way, the row address and write data are applied, and as described above, the memory access mode signal of line L1 is at "1," so the memory control ring 6 commences operations, and there are state transitions so that outputs T0, T1, . . . Tk sequentially go to "1." When Tk goes to "1." the "1" output Tk is applied to terminal D of the RAS flipflop 7 through the jumper line 17 and OR gate 11, and when output Tk + 1 of memory control ring 6 goes to "1," the RAS signal that is output from output terminal 1 of the RAS flipflop 7 goes to "1." In addition, at this time the inverted signal that is output from the output terminal 0 of the RAS flipflop 7 goes to "0," causing the output of the NOR gate 14 to go to "1," and the output of the OR gate 11 to go to "1," causing the output of the RAS flipflop 7, or in other words the RAS signal, to be held at "1" even if the memory control ring 6 status advances. When the "1" output of the memory control ring 6 transitions from Tl to Tl + 1, the same operation as described above causes the COLS signal, which is the output of the COLS flipflop 9, to be held at "1." This COLS signal causes the address multiplexer 2 to output the column address to the multiplexed address bus 3, and the output of the AND gate 10, or in other words the WE signal that is applied to the RAM 1 terminal WE, goes to "1," placing RAM 1 in write mode. When the "1" output of the memory control ring 6 transitions from Tm to Tm + 1, a operation similar to what was described above causes the output of the CAS flipflop 8, or in other words the CAS signal, to be held at "1." As described above, the RAS signal, the CAS signal, the WE signal, and the COLS signal all go to "1," putting all conditions in place to write to the RAM 1; hence the data write operation is performed, the status of the memory control ring 6 advances, and the write operation is concluded at the point in time where the output Tn - 1 goes to "1." When the output Tn of the memory control ring 6, or in other words the memory access complete signal on line 4, goes to "1" followed by the output Tn + 1 going to "1," the memory access mode signal on line L1 and the write mode signal on line L3 both go to "0," causing the outputs of the NOR gates 14, 15, and 16, along with the outputs of the OR gates 11, 12, and 13 to go to "0"; consequently, the RAS signal, the CAS signal, and the COLS signal all go to "0," completing the operation for writing to the RAM 1. Note that TW shown in Figure 4 is the period over which the write mode conditions are fulfilled by the control signals to the RAM 1 (i.e., the memory access mode signal, the write mode signal, the RAS signal, the COLS signal, the CAS signal, and the WE signal).

On the other hand, in the operations to read from the RAM 1, as shown in Figure 5, the write mode signal and the WE signal go to "0," and at the point in time when the output Tn - 1 of the memory control ring 6 ceases to output "1," or in other words, at the point in time when the output Tn goes to "1," the output data that is read from the RAM 1 is assumed to be set, and with the output Tn, the data on the data bus 5 is accepted. At this time, when, in operations similar to the write operations described above the output Tn + 1 of the memory control ring 6 is to go to "1," all control signals become inactive and the operations to read from the

RAM 1 are terminated. Note that the TR shown in Figure 5 is the period of time over which the control signals to the RAM 1 fulfill the read mode conditions.

[Problems Solved by this Invention]

In the conventional method for setting the memory access timing, the part that sets the RAM access timing is set by jumper lines, and thus it requires a manual intervention to set the jumper lines. Additionally, generally RAMs have a variety of different access times, and when the type of RAM that is used is changed it is necessary to change the settings of the jumpers in order to change the access timing, and, as a result, the RAM cannot be accessed correctly if the setting is incorrect or there may also be the problem that, even if RAM that can operate at high speeds is used, the actual performance of the RAM will not be good if the access timing used is for low speed RAM.

This invention was created in order to solve the types of problems described above, and its objective is to provide a method of setting the memory access timing that automatically sets the access timing without any manual intervention, making it possible to exploit the full capabilities of the RAM and to improve reliability.

[Method by Which the Problems are Solved]

The method of setting the memory access timing in this invention is characterized by the logic circuits that access the memory (RAM 1) being equipped with registers 21, 22, 23, and 24 that can be set to a variety of values by a program, where the values that are set to these registers 21, 22, 23, and 24 are repetitively changed sequentially by the program at which time test data is written to and read from the memory (RAM 1) and comparisons are made between the write data and the read data where the values that were set when the results of the comparison indicates a match are set to registers 21, 22, 23, and 24, so that the access to the memory (RAM 1) is performed based on these settings.

[Operation]

The registers 21, 22, 23, and 24 in this invention are set to any given value by the program, and the memory (RAM 1) is accessed based on the various settings that have been set, at which time test data is written to the memory and read from the memory. The data written as this test data, and the data that is read, are compared to each other for each of the access operations that are based on the respective settings, and when the data that is written to the memory (RAM 1) matches the data that is read from the memory, then the settings are set as the final settings in the registers 21, 22, 23, and 24, and after that time the access timing is determined based on these final settings and the memory (RAM 1) is accessed with that access timing when the specific data write and data read operations are performed.

[Example of Embodiment]

An Example of Embodiment of this invention is explained below based on the figures. Figure 1 is a block diagram of logic circuits that use the method for setting the memory access timing in this Embodiment of the invention. In Figure 1 the same symbols are used as corresponding to the structural elements shown in Figure 6, so the explanations are omitted here. In Figure 1, 21 is the register for determining the timing with which the RAS signal is produced (hereinafter termed the "RAS register"), 22 is the register for determining the timing with which the COLS signal is produced (hereinafter termed the "COLS register"), 23 is the register for determining the timing with which the CAS signal is produced (hereinafter termed the "CAS register"), 24 is the register for determining the timing with which the memory access complete signal will be produced (hereinafter termed the "CPLT register"), 25, 26, 27, and 28 are the selectors that select one output from output T0 to Te of the e + 1 registers in memory control ring 6.

Next the operation will be explained. Let us assume that there are five different types of RAM that can be obtained, and, the access timing on these types of RAM, from fastest to slowest, are RAM_1, RAM_2, RAM_3 RAM_4, and RAM_5. The respective RAMs can be accessed correctly by outputting the RAS signals, COLS signals, CAS signals, and memory access complete signals shown in the timing diagram of Figure 2. The explanation described below considers the operations when RAM_2 is installed.

The table has the settings for the RAS register 21, the COLS register 22, the CAS register 23, and the CPLT register 24, or in other words, the settings for k1 to k5, l1 to l5, m1 to m5, and n1 to n5 in Figure 2, are stored as a table. This program executes the flow chart shown in Figure 3. In other words, the program is executed (Step S1), the pointer indicates RAM_1 (Step S2), the information indicated by the pointer (in this case, the settings k1 corresponding to RAM_1 shown in Figure 2) are loaded into RAS register 21 (Step S3), the pointer is then incremented (Step S4), the information indicated by the pointer (in this case, the setting l1 corresponding to RAM_1) is loaded into the COLS register 22 (Step S5), the pointer is incremented (Step S6), the information indicated by the pointer (in this case, the setting m1 corresponding to RAM_1) is loaded into the CAS register 23 (Step S7), the pointer is incremented (Step S8), the information indicated by the pointer (in this case the setting n1 corresponding to RAM_1) is loaded into the CPLT register 24 (Step S9), the pointer is incremented (Step S10), the test data is written into the RAM_2 (because in this case it is RAM_2 that is installed) (Step S11), and the write operation is performed with the timing shown in Figure 4. Then the data is read from the RAM_2 with the timing shown in Figure 5 (Step S12), and the data that was read is compared to the data that was written (Step S13). In this case, the settings are the settings k1, l1, m1, and n1 that correspond to RAM_1. These settings do not match the timing for the control signals (the RAS signal, the COLS signal, the CAS signal, and the memory access complete signal) for RAM_2, so the comparison in Step 13 of the data that was read and the data that

was written does not indicate a match with this timing. As a result, the program continues to Step S14, and a check is made for a pointer error. If there is an error then an error report is made (Step S15), and if there is no error, then the program returns to Step S3.

The information indicated by the pointer when the program returns to Step S3 is the setting k2 that corresponds to the RAM_2 that is installed, and this setting k2 is loaded into the RAS register 21. After that, the same process that is described above is performed (Steps S4 through S10) and the setting l2 is loaded into the COLS register 22, the setting m2 is loaded into the CAS register 23, the setting n2 is loaded into the CPLT register 24, the test data is written to the RAM_2 (Step S11) the data is read from the RAM_2 (Step S12), and the data that was written is compared to the data that was read (Step 13). In this case, the RAM access timing is set so that, when the operations for writing and reading the specified data are performed, the settings k2, l2, m2, and n2 correspond to RAM_2, and thus RAM_2 is accessed with the appropriate timing and the data that was read matches the data that was written so the program continues to Step 16 and the settings k2, g2, m2, and n2 are set into registers 21, 22, 23, and 24 as the final settings, and selectors 25, 26, 27, and 28 cause the RAS signal to be "1" when the output Tk2 + 1 of the memory control ring 6 is "1," the COLS signal to be "1" when the output Tl2 + 1 is "1," the CAS signal to be "1" when the output Tm2 + 1 is "1," and the memory access complete signal to be "1" when the output Tn2 is "1." In addition, when the output Tn2 + 1 is "1" the RAS signal, the COLS signal, the CAS signal and the memory access complete signal all go to "0."

While the explanation of the flow chart was based on the assumption that RAM 2 was installed, if RAM_1, RAM_3, RAM_4, or RAM_5 were installed instead, the processes in Steps 3 through 13 would be performed once, three times, four times, or five times, respectively, to set the access timing.

Because in the Example of Embodiment described above, it is possible to change the access timing using a program, it is easy to perform RAM access timing margin tests. In addition, although the timing will be that for the type of RAM with the slowest access time, even if a mixture of RAMs with different access times are installed in the logic circuits, the RAM can still be accessed correctly. In addition, if in high-speed computers, the RAM access timing is set individually by the card unit or the bank unit of main memory, then even if the type of RAM is different on different card units or bank units, the timing can be performed to match the capability of the RAM, making it possible to prevent any impediments to performance by mixing types of RAM. Additionally, in the program that determines the settings, it is possible to set the access timing that is optimized for the RAM that is installed and that is able to fully exploit the capabilities of the RAM through selecting the optimal values through changing the settings in even finer increments, rather than determining the settings in such a way as to compensate for the minor timing differences between the various RAM manufacturing locations. If in the program access timing setting checks are

performed for all addresses of all RAM, then it is possible to identify the RAM that has errors even if different types of RAM (with different access times) are mixed.

Furthermore, in the Example of Embodiment described above, dynamic RAM was used as the example, when static RAM is used then the chip select (CS) signal and the output enable (OE) signal can be controlled instead of the RAS signal and the CAS signal. Although in the Example of Embodiment above a memory control ring was used to control the RAM access timing, the method of this invention can also be performed by establishing for each signal to be controlled by the program a combination of a counter into which data can be loaded and a register that sets the value that is loaded into the counter as the initial value.

[Effects of the Invention]

Using the invention described above, it is possible to set the memory access timing automatically without a manual intervention because a register is provided wherein a variety of different values can be set by the program where the values that are set into this register are repetitively changed sequentially, test data is written to the memory and then read from the memory, and the data that was written is compared to the data that is read and the values that were set when the results of the comparison indicate a match are set to the register so that the memory is accessed based on those settings, it is able to prevent any disruptions to memory performance or situations where access cannot be performed normally due to incorrect settings in the access timing, making it possible to fully exploit the capabilities of the memory, and thus possible to obtain the effect of increased reliability; in addition, it is no longer necessary to have a manual intervention in order to set the access timing using jumper lines as it has been conventionally, thus making it possible to reduce operating test expenses and reduce labor expenses, and making it possible to provide data processing equipment less expensively.

4. Simple Explanation of Figures

Figure 1 is a block diagram of the logic circuits that use the method for setting the memory access timing in the Example of Embodiment of this invention.
Figure 2 is a timing diagram showing the relationship between the settings and the access timing in this Example of Embodiment.
Figure 3 is a flow chart used for explaining the operation of the Example of Embodiment.
Figure 4 is a timing chart for explaining a conventional example and explaining the operations for writing to the RAM in the Example of Embodiment thereof.
Figure 5 is a timing chart for explaining the conventional example and for explaining the operations for reading from the RAM in an example thereof.
Figure 6 is a block diagram of the logic circuits that use the conventional method of setting the memory access timing.

1: RAM (memory)
2: Address multiplexer
6: Memory control ring
7: RAS flipflop
8: CAS flipflop
9: COLS flipflop
10: AND gate
11, 12, 13: OR gates
14, 15, 16: NOR gates
21: RAS register
22: COLS register
23: CAS register
24: CPLT register
25, 26, 27, 28: Selectors

Representative: Masao Oiwa, and two others

Figure 1
[L1] Memory access mode signal
[L2] Fundamental clock
[6] Memory control ring
[21] Register
[25] Selector
[22] Register
[26] Selector
[23] Register
[27] Selector
[24] Register
[28] Selector
[L4] Memory access complete signal
[Under 14] RAS signal
CAS signal
[2] Address multiplexer
[Above 10] COLS signal
[L3] Write mode signal

Figure 2
[INSERT TABLE]

Type of RAM	RAS signal timing	COLS signal timing	CAS signal timing	Memory access complete signal timing	Settings for the registers for generating the timing			
					Registe	Registe	Registe	Registe

HKR185566

HR905_066528

			r 21	r 22	r 23	r 24
[see source for English]						

Figure 3

S1 Start
S2 Set pointer = PRAM 1
S3 Load the information indicated by the pointer into the RAS register
S4 Pointer = pointer + 1
S5 Load the information indicated by the pointer into the COLS register
S6 Pointer = pointer + 1
S7 Load the information indicated by the pointer into the CAS register
S8 Pointer = pointer + 1
S9 Load the information indicated by the pointer into the CPLT register
S10 Pointer = pointer + 1
S11 Write test data to the RAM
S12 Read test data from the RAM
S13 Compare the read data to the write data
S14 Is there a pointer error?
S15 Error report
S16 Settings complete

Figure 4

Fundamental clock
Memory access mode signal
Write mode signal
T0
Tl
Tk
Te
Tm
Tn (Memory access complete signal)
RAS signal
COLS signal
CAS signal
WE signal

Figure 5

Fundamental clock
Memory access mode signal
Write mode signal
T0
Tl
Tk
Te
Tm
Tn (Memory access complete signal)

HKR185567

HR905_066529

RAS signal
COLS signal
CAS signal
WE signal

Figure 6
[L1] Memory access mode signal
[L2] Fundamental clock
[6] Memory control ring
[L4] Memory access complete signal
[Under L4] RAS signal
CAS signal
WE signal
[2] Address multiplexer
[Above 10] COLS signal
[L3] Write mode signal